- Chollet, François. "On the Measure of Intelligence.” (2019).
- https://arxiv.org/abs/1911.01547

Slides created by Yegon Kim
2025/01/08

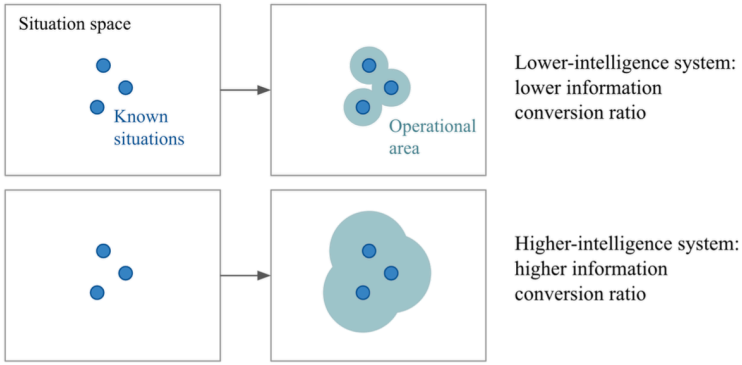
Motivation

- What are intelligent systems?
- e.g. Is ResNet18 trained on CIFAR-10 an *intelligent* system?

  What about OpenAI o1?
- *Objective: define intelligence and propose a new benchmark*

Two Conceptions of Intelligence

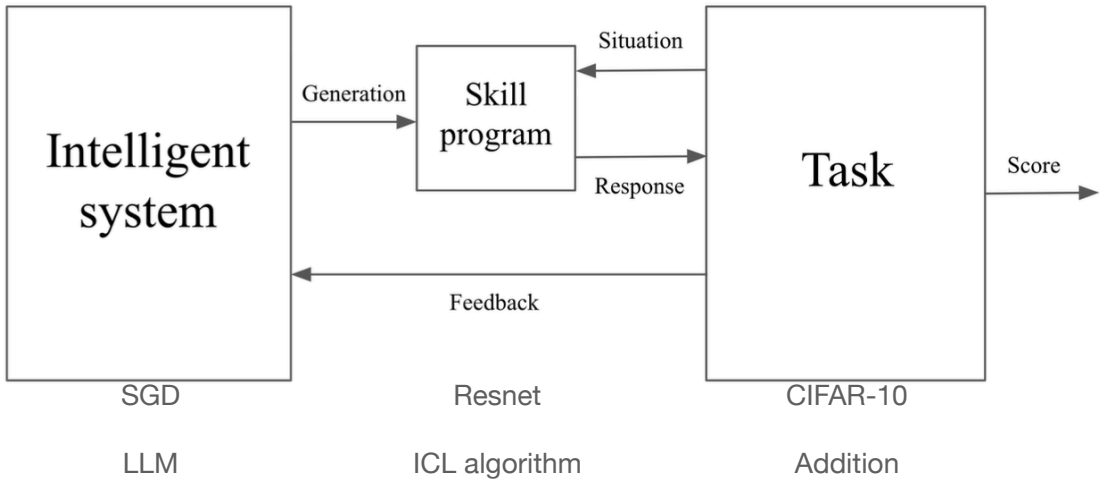
- Intelligence as Task-Specific Skill
- *Intelligence as General Learning Ability*



- One can simply ‘buy’ a skill by putting in a lot of data (memorization)
- But doing so efficiently is intelligence (Chollet)

Proposed Definition

- Intelligent system outputs a skill program according to Task
- Skill program interacts with Task



Proposed Definition

- Prior: how much information does an IS have about a Task at t=0

$$P^{\theta}_{IS,T} = \frac{H(Sol^{\theta}_{T}) - H(Sol^{\theta}_{T}|IS_{t=0})}{H(Sol^{\theta}_{T})}$$

- Experience: how much information is accrued at each timestep

$$E^{\theta}_{IS,T,t} = H(Sol^{\theta}_{T}|IS_{t}) - H(Sol^{\theta}_{T}|IS_{t}, data_{t})$$

- Generalization difficulty: how much does the optimal training-time solution generalize at test time (~distribution shift)

$$GD^{\theta}_{T,C} = \frac{H(Sol^{\theta}_{T}|TrainSol^{opt}_{T,C})}{H(Sol^{\theta}_{T})}$$

Proposed Definition

- Intelligence as *Skill-Acquisition Efficiency*
- “The intelligence of a system is a measure of its *skill-acquisition efficiency* over a scope of tasks, with respect to *priors, experience, and generalization difficulty*.”

$$I^{\theta_T}_{IS,scope} = \underset{T \in scope}{Avg} \left[\omega_T \cdot \theta_T \sum_{C \in Cur_T^{\theta_T}} \left[P_C \cdot \frac{GD^{\theta_T}_{IS,T,C}}{P^{\theta_T}_{IS,T} + E^{\theta_T}_{IS,T,C}} \right] \right]$$

Intelligence

Tasks

Value of solving the task

Sufficient score

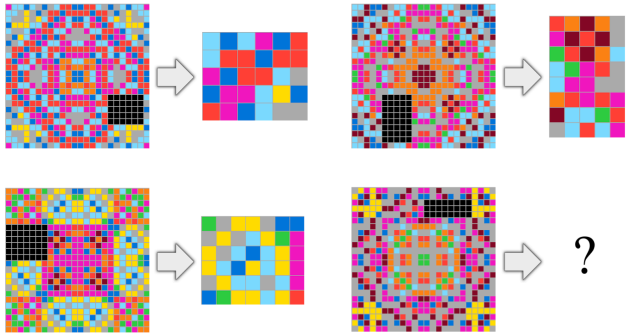
Curricula

P(Curriculum)

Generalization difficulty / (Prior + Experience)

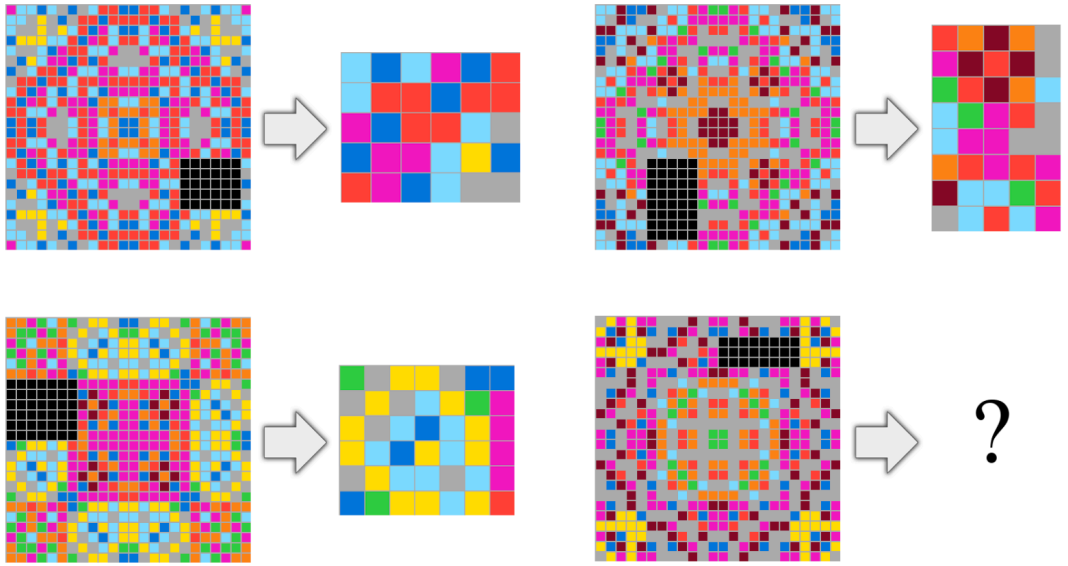
Benchmark

- Each problem is a distinct task
- Should be solvable by humans without training
- Focus on developer-aware generalization (unseen tasks in test set)
- Focus on broad generalization, requiring only a few examples



ARC Benchmark

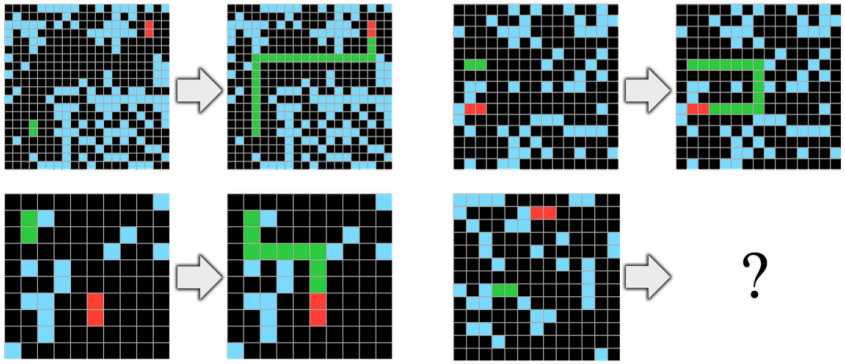
- Abstraction and Reasoning Corpus
- 2D grids
- 3~4 input-output pairs
- 1~2 test inputs



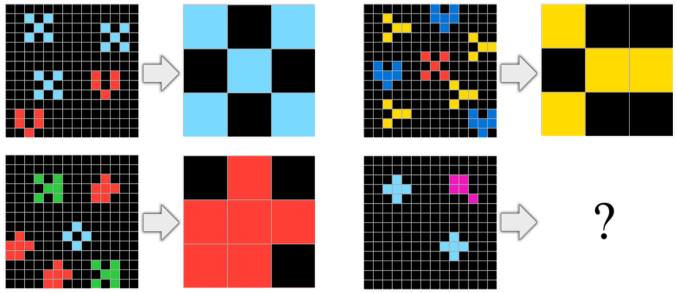
Inferring hidden grids through symmetry

Core knowledge priors

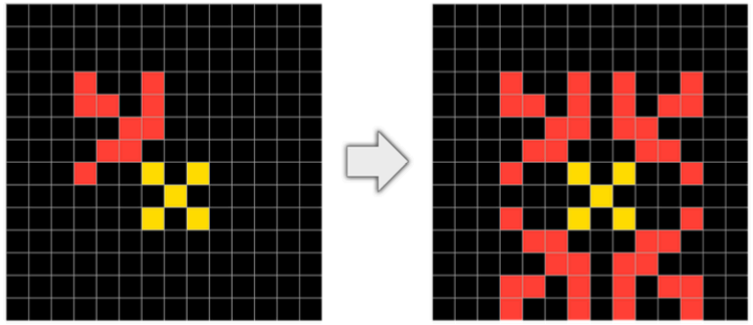
- Objectness prior
- Goal-directedness prior
- Numbers and Counting priors
- Basic Geometry and Topology priors



Goal-directedness prior



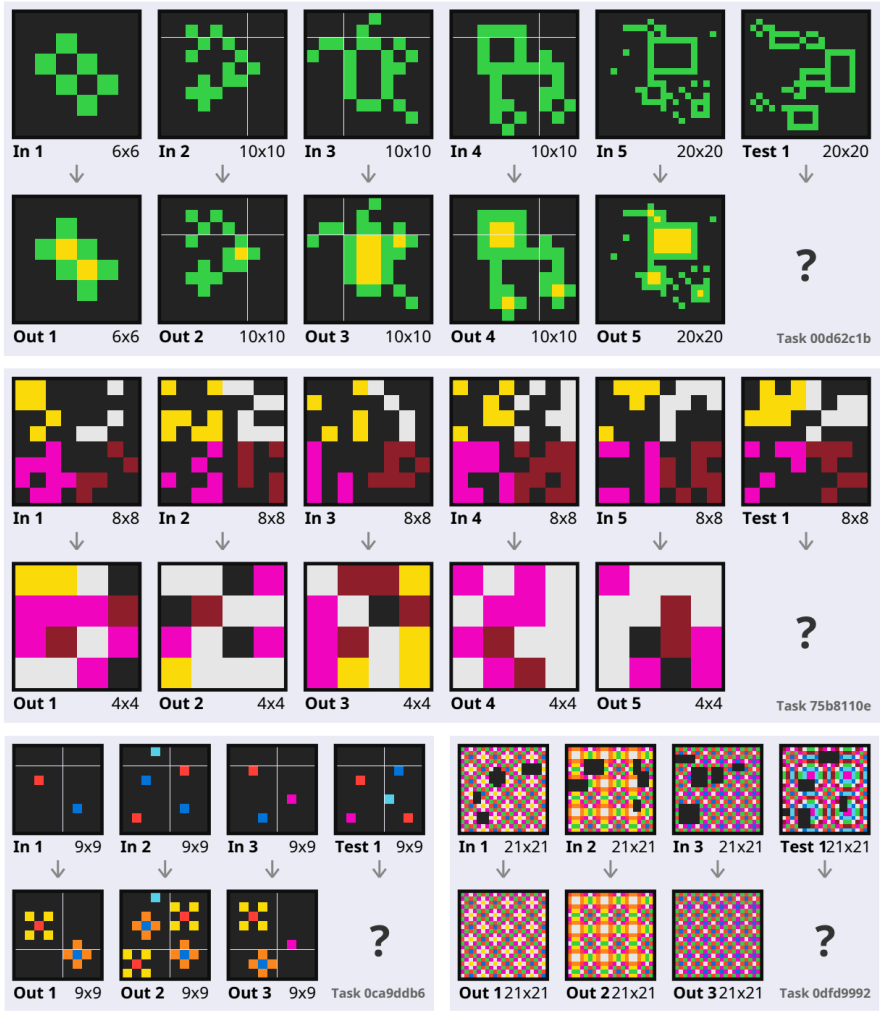
Numbers and Counting priors



Basic Geometry and Topology priors

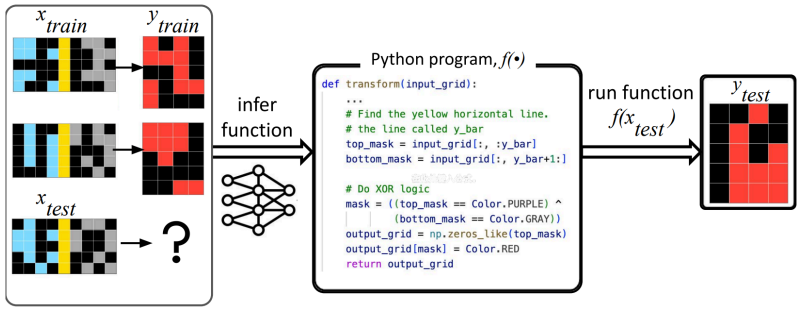
Examples

- 400 in training set
- 400 in public eval set
- 100 in private eval set
  Good human gets ~85% (?)
  Average gets ~75%
- $1,000,000 prize for >85% (2024~)
- Eval set is created with *different program templates*



Chollet's idea

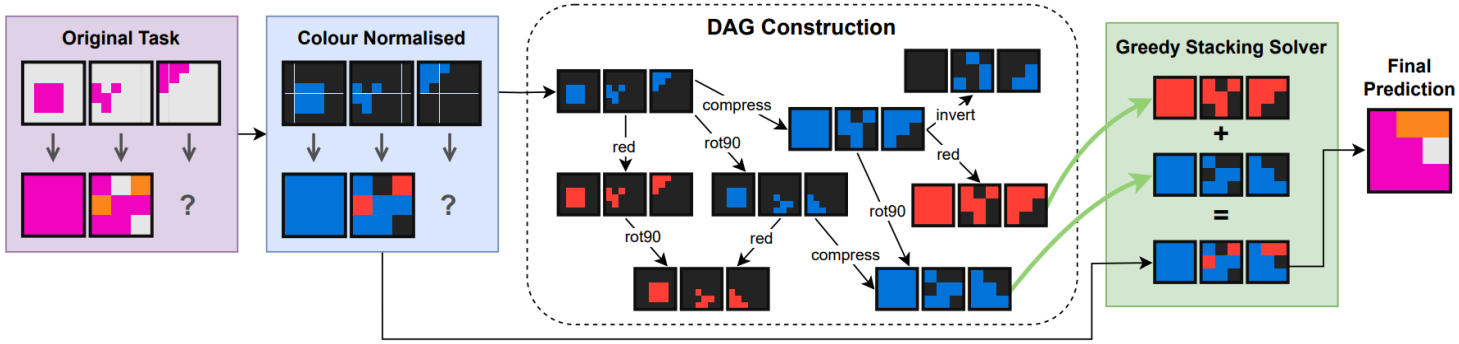
- Neural-network guided program synthesis



- Why not:
  - SGD: requires too much data (ARC has 3~4 examples per task), no guarantee on complexity, etc.
  - Meta-learning: can only retrieve programs seen during training (why?), also no guarantee on complexity, etc.

Existing approaches

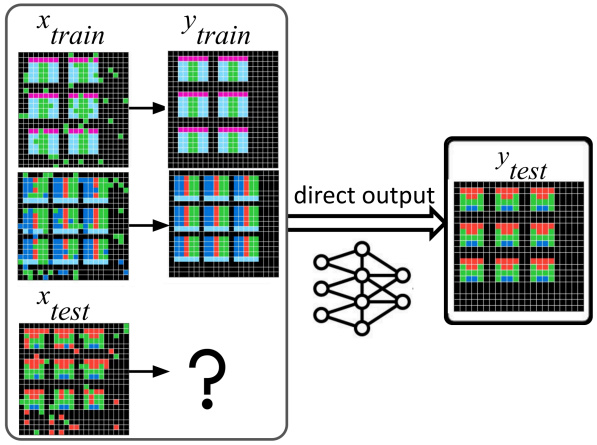
- Program synthesis with DSL (domain-specific language)
- e.g.
  Icecuber — 2020 winner, BFS search, 142 hand-crafted primitives
  DreamCoder — neural network guided search, learns new abstract primitives



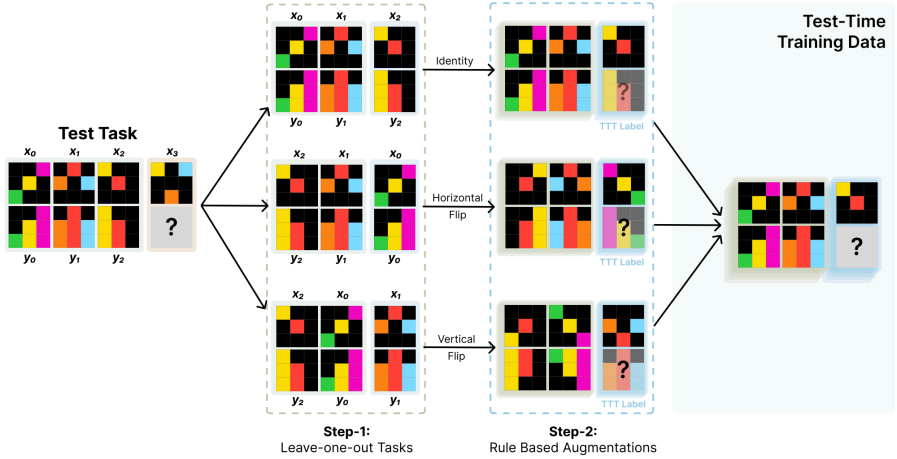
Icecuber

Existing approaches

- Meta-learning + test-time augmentation
- e.g. Jack Cole et al. (2023, 2024 winner) ~ 39% (June 2024) -> 55% (Dec 2024)
- https://lab42.global/community-model-efficiency/



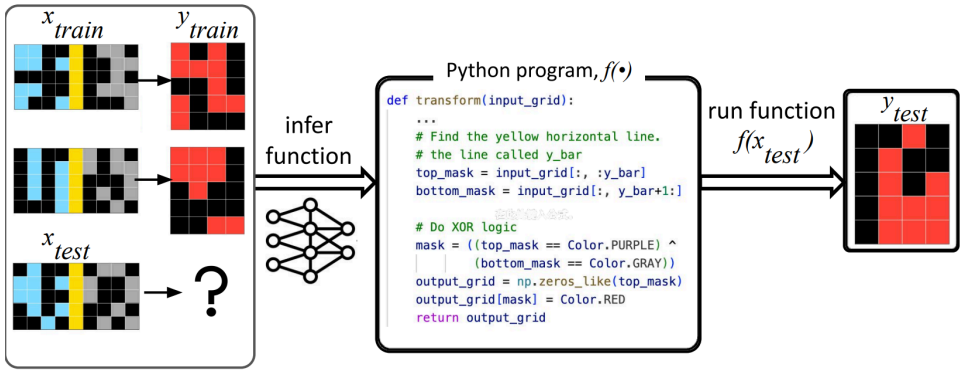
Meta-learning

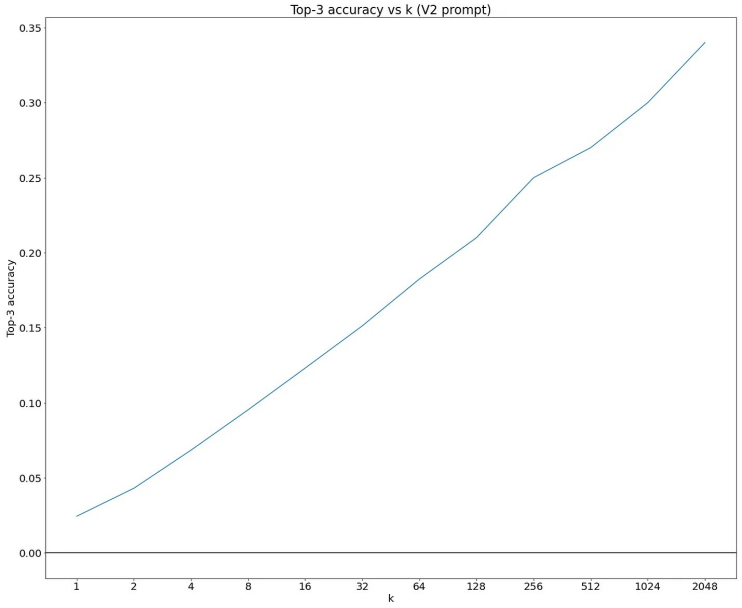


Test-time augmentation

Existing approaches

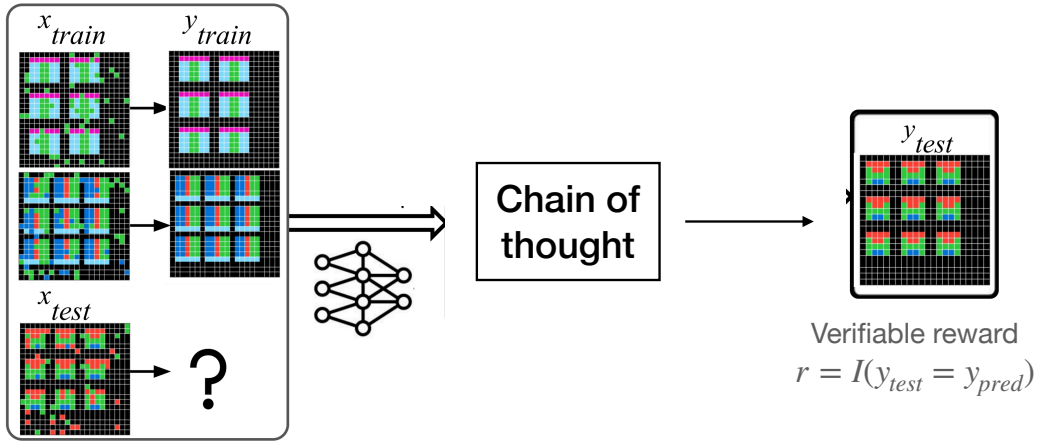
- Program synthesis with python, using LLM
- Generate 1000 candidate programs, find one that satisfy the observed input-output pairs, apply it to the test input
- e.g. Ryan Greenblatt with GPT-4o

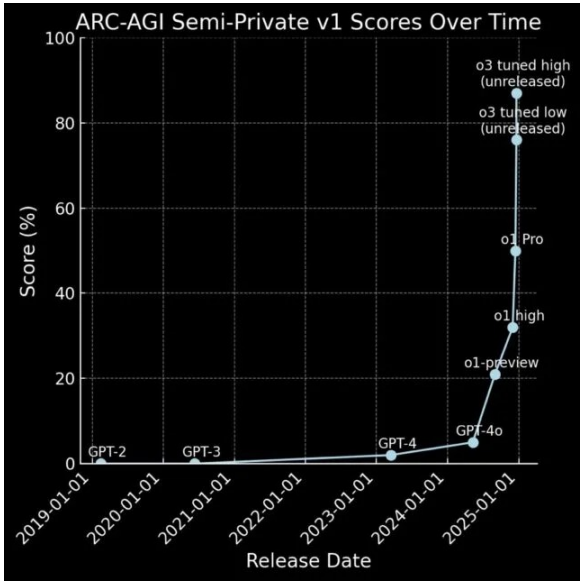




Existing approaches

- Just prompt LLMs
- o3 trained (RL) only on public training set achieves ~87%





Extra Thoughts

- Slight nitpick at the definition: $$I^{\theta_T}_{IS,scope} = \underset{T \in scope}{Avg}\left[\omega_T \cdot \theta_T \sum_{C \in Cur^{\theta_T}_T}\left[P_C \cdot \frac{GD^{\theta_T}_{IS,T,C}}{P^{\theta_T}_{IS,T}+E^{\theta_T}_{IS,T,C}}\right]\right]$$
  - Value should be all there is to the weighting.
    e.g. It should contain the usefulness of generalizing w/ small experience

$$I = Avg_T\,[\sum_C [P_C \cdot \omega_{T,C}]]$$

- “Understanding” is the act of decomposing a phenomenon into simpler components
- LLM program search is still wildly expensive

Extra Thoughts